REMARKS

Reconsideration and allowance of the subject application are respectfully requested

As a preliminary matter, the Examiner is again requested that the information disclosure papers filed on November 8, 2002 be considered and an initial copy of the PTO-1449 form submitted with that Information Disclosure Statement be returned.

Applicants appreciate the courtesies extended by Examiner Vu and Khatri during the interview conducted on May 4, 2005. The final rejection based on the garbage collection system of Alpern was discussed in contrast with the supervision methodology represented by the instant claims. At the encouragement of Examiners Vu and Khatri, Applicants have amended the claims to emphasize these distinctions.

Claim 1 now recites a method of supervising the execution of one or more program sections is to "detect an object that unexpectedly disrupts execution in one or more program sections." See page 8, lines 18-20 of the instant specification. The storing step has been amended to include "the one or more information units associated with the created object allowing supervision of execution of the program section." As was explained during the interview, garbage collection is a technique for automatically handling memory deallocation, and it triggers on whether an object is "accessible" or not, i.e., if there exists any path between a root set and an object, as explained in column 2, lines 32-46. When developing software a system that provides garbage collection, a programmer need not keep track of memory areas hosting the objects. Any object that cannot be accessed any longer is removed. But as long as objects can be accessed, the garbage collection ignores them--regardless of whether those objects are used by the program any longer. As a result, a garbage collector cannot detect if or when an accessible object ceases to be active with respect to its purpose in the program. Only when the garbage collector deems that an object has no root path, and therefore cannot be assessed, does it remove the object and return the memory space to the memory pool.

In contrast to garbage collection, the instant claims relate to program execution surveillance. Indeed, that program execution surveillance could be used along with garbage collection, but it is separate from garbage collection. In many programs, there is often a subset of objects that should not be allowed to become "lost" or enter into a lost state waiting for external events that do not occur. Such lost objects are still accessible with respect to garbage collection, but nevertheless something has failed to operate properly requiring some kind of alarm or notification to an operator. In essence, a lost object has become non-responsive and has failed to perform its purpose. In contrast with garbage collection in which an inaccessible object is removed, the claimed method/apparatus retains a lost object even after sending out an alarm signal.

Thus, the claims focus on identifying disruptions or failures in the program execution by scanning for information units that have been stored too long and which would otherwise has been removed during normal operation. As explained in step (d) in claim 11, the memory is scanned to "identify one or more information units associated with an object that is not completed and for which there has been no activity for a time period longer than the associated expiration time period." Such scanning and identification is simply not performed in Alpern. Moreover, Alpern does not disclose for such identified information units, "triggering an alarm signal to indicate that an unexpected disruption of execution has been detected."

The Examiner admits that Alpern does not disclose triggering an alarm signal, but relies on "the concept" of generating an alarm when a "violation in memory or execution time reference conflict is encountered." Applicants are not claiming to be the first to use an alarm or an interrupt when some general kind of "violation" has occurred. Nor are the independent claims that general. Indeed, an alarm is not generated simply when a violation occurs, but rather, only after scanning the memory identifies an information unit associated with an object that is not completed and for which there has been no activity for a time period longer than the associated expiration time period. When such an identification occurs, then the alarm signal triggered. And again, the alarm signal has nothing to do with garbage collection. Rather, the alarm signal indicates "that an unexpected disruption of execution has been detected."

Even if one were to combine the teachings of Nilsen, which describes at column 14, lines 20-25 an interrupt signal to a microcontroller 23 to "fix up" an invalid data transfer with Alpern's garbage collection scheme, the result would be sending an alarm to an operator many times per second! A person of ordinary skill in the art would certainly not have been motivated to create and send so many alarm signals per second in Alpern because that would paralyze the memory management system.

The application is in condition for allowance. An early notice to that effect is earnestly solicited.

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: ______________________
John R. Lastova
Reg. No. 33,149

JRL:sd
901 North Glebe Road, 11th Floor
Arlington, VA 22203-1808
Telephone: (703) 816-4000
Facsimile: (703) 816-4100